On-board Battery Monitoring and Prognostics for Electric-Propulsion Aircraft

Chetan Kulkarni*
Stinger Ghaffarian Technologies Inc., NASA Ames Research Center, Moffett Field, CA, 94035, USA

Johann Schumann†
Stinger Ghaffarian Technologies Inc., NASA Ames Research Center, Moffett Field, CA, 94035, USA

Indranil Roychoudhury‡
Stinger Ghaffarian Technologies Inc., NASA Ames Research Center, Moffett Field, CA, 94035, USA

The reliability of the propulsion system of an aircraft is paramount for the aircraft safety and hence the aircraft health must be monitored continuously. In contrast to fuel-operated aircraft, electric battery-operated propulsion system poses specific problems, such as, the remaining battery power does not linearly decrease and cannot be measured directly. In this paper, we describe a combined monitoring and prognostics architecture that can continuously monitor all components of the electric propulsion system with respect to safety and performance properties as well as state of charge and rest of useful life for the battery. Our system combines a detailed electrochemical battery model for Li-ion batteries with a powerful prognostics engine based upon an Unscented Kalman Filter with the R2U2 monitoring device, which provides efficient observers for metric temporal logic and Bayesian reasoning. R2U2 is a real-time, REALIZABLE, RESPONSIVE, UNOBTRUSIVE UNIT, **which continuously monitors sensor readings, outputs of the prognostics engine, as well as the flight software status for safety, performance, and security properties. We illustrate our architecture with two case studies, one reporting actual flight tests with an X8+ octo-copter and the other a software-in-the-loop simulation with an unmanned Edge 540 electric aircraft model.**

I. Introduction

The reliable operation of the propulsion system of an aircraft is vital for its safety. As a result, the health of the propulsion system must be monitored continuously and failures diagnosed immediately. There also must be means to, at any point in time during the flight, reliably estimate if enough resources (gas, battery capacity) are available to safely conclude the flight, even under failure conditions. In traditional gas-powered aircraft, the pre-flight calculation of the necessary fuel, and ample fuel reserves are prerequisites for a safe flight. The pilots are trained to carefully watch the operation of the engines and the fuel consumption and estimate the remaining fuel during the flight based on their expertise and experience. Yet there have been numerous incidents where empty fuel tanks led to dangerous situations or accidents.[1]

On an aircraft with battery-powered electric propulsion, the situation is much more complicated. The remaining time the aircraft can continue its flight depends on the current and future capacity of the on-board batteries. That battery capacity cannot be measured directly and depends, in a highly non-linear way, on numerous factors such as, the age of the battery, number of loading cycles, temperature, and drawn current. This makes a simple "battery gauge" impossible. In addition, the energy density (kWh/kg) of state-of-the-art batteries is comparatively low. Thus carrying ample battery reserves on board is not possible due to their prohibitive weight. Therefore, the current battery capacity and remaining available flight time must be

*SGT Inc., Intelligent Systems Division, NASA Ames Research Center, chetan.s.kulkarni@nasa.gov
†SGT Inc., Intelligent Systems Division, NASA Ames Research Center, johann.schumann@nasa.gov
‡SGT Inc., Intelligent Systems Division, NASA Ames Research Center, indranil.roychoudhury@nasa.gov

estimated with high accuracy and confidence throughout the entire flight and the entire propulsion system must be monitored with respect to safety and performance properties on-board and in real time.

To this end, in this paper, we present (a) a high-fidelity battery performance model that is based upon electro-chemical reactions, (b) a powerful prognostics engine, which uses this model to provide, in real-time, reliable estimates of the state-of-charge (SoC) of the battery as well as it rest of useful life (RUL) under various load conditions, and (c) a monitoring framework (R2U2) for the on-board monitoring of the aircraft's propulsion system, continuous checking of safety properties, as well as on-board diagnosis and probabilistic root cause analysis.

Our integrated framework uses state-of-the-art technology like Unscented Kalman Filters (UKF), monitoring with temporal logic observers, and Bayesian reasoning. Its output comprises safety status of the aircraft and predicted energy capacities and can also include a measure on the estimates' confidence. This information can be provided to the pilot of the electric aircraft using a suitable display and annunciation (not part of this paper). For the operation of an unmanned aerial system (UAS), the output of our prognostics and monitoring system can be directly fed into the on-board autopilot system.

The rest of the paper is structured as follows: Section II discusses related work on prognostics and system health management. In Section III we present our electro-chemical battery model. Our prognostics engine will be discussed in Section IV; Section V provides an overview of our monitoring framework and shows some examples of health management properties. In Section VI we present our case studies and Section VII discusses future work and concludes.

II. Related Work

Previous work on battery health monitoring introduced several tools for battery discharge prediction onboard an electric aircraft. Work by Bole et. al.[2] discusses a battery equivalent circuit model to simulate battery health state. Current and voltage profiles logged during flights of a small electric airplane further tuned the battery model.[2] The implementation of a flight plan with upper and lower uncertainty bounds on the required energy consumption to complete the mission successfully is presented by Quach et. al.[3] along with an approach to identify additional parasitic battery loads. A verification testing procedure intended to build trust in predictions of remaining flying time prior to actual flight testing is presented with results from several flights and the verification testing of remaining flying time prior to flight testing is discussed in earlier publications.[3,4] The key idea behind these earlier works is to demonstrate the translation of system performance goals and safety requirements into an warning system that indicates the operator when the estimated remaining flying time falls below a certain threshold. Hogge et. al.[5] demonstrated some of the variation the powertrain and the pilot may introduce while verifying and analyzing any risks before the actual flight. This was accomplished through several vehicle ground tests, which provides the closest possible testing conditions to an actual flight.

There is a wealth of temporal-logic runtime monitoring techniques in software, including automata-based, low-overhead techniques,[6,7] Copilot,[8] or BusMOP.[9,10] These systems focus on property monitoring using variants of temporal logic, but do not combine monitoring with Bayesian probabilistic reasoning or with a prognostics engine. The integration of R2U2 with a prognostics engine and Bayesian reasoning is described in an earlier paper.[11]

III. Battery Modeling

In order to predict end-of-discharge (EOD) as defined by a voltage cutoff, the battery model must compute the voltage as a function of time given the current drawn from the battery. There are several electro-chemical processes that contribute to the cell's potential that make this a difficult problem. For the purposes of on-line prognostics, we focus here on a lumped-parameter ordinary differential equations form that still considers the main electro-chemical processes.

The voltages of a single cell in a battery pack are summarized in Figure 1 (adapted from the paper by Rahn and Wang[12]). The overall battery voltage $V(t)$ is the difference between the potential at the positive current collector, $\phi_s(0,t)$, and the negative current collector, $\phi_s(L,t)$, minus resistance losses at the current collectors (not shown in the diagram). As shown in the figure, the potentials vary with the distance $d \in [0, L]$, because the loss varies with distance from the current collectors. Details of the developed battery model are discussed in the paper by Daigle & Kulkarni.[13]

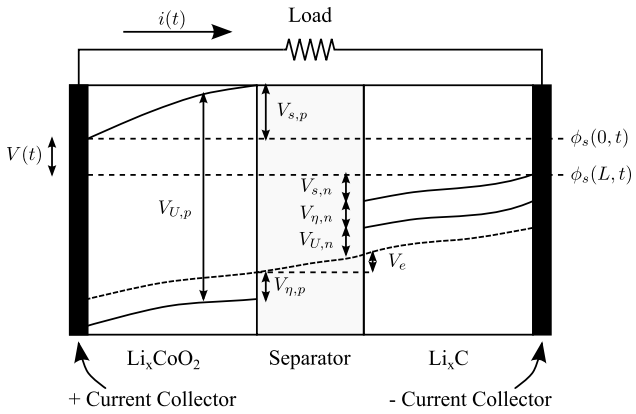


Figure 1. Battery voltages.

III.A. State of Charge

State of Charge (SOC) of a battery is conventionally defined to be 1 when the battery is fully charged and 0 when the battery is discharged to a set voltage threshold. In this model, it is analogous to the mole fraction x_n, but scaled from 0 to 1. There is a difference here between nominal SOC and *apparent* SOC. Nominal SOC would be computed based on the combination of the bulk and surface layer control volumes in the negative electrode, whereas apparent SOC would be computed based only on the surface layer. That is, a battery can be discharged at a given rate, and reach the voltage cutoff, i.e., apparent SOC is then 0 as discussed earlier. But, once the concentration gradient settles out, the surface layer will be partially replenished and the battery can be discharged further, i.e, apparent SOC increases whereas nominal SOC remains the same.

Nominal (n) and apparent (a) SOC are defined by equations below:

$$SOC_n = \frac{q_n}{0.6q^{\max}}, \text{and} \tag{1}$$

$$SOC_a = \frac{q_{s,n}}{0.6q^{\max_{s,n}}}, \tag{2}$$

where $q^{\max_{s,n}} = q^{\max}\frac{v_{s,n}}{v_n}$. The factor 1/0.6 comes from the fact that the mole fraction at the positive electrode cannot go below 0.4,[13] therefore SOC of 1 corresponds to the point where $q_n = 0.6q^{\max_{s,n}}$.

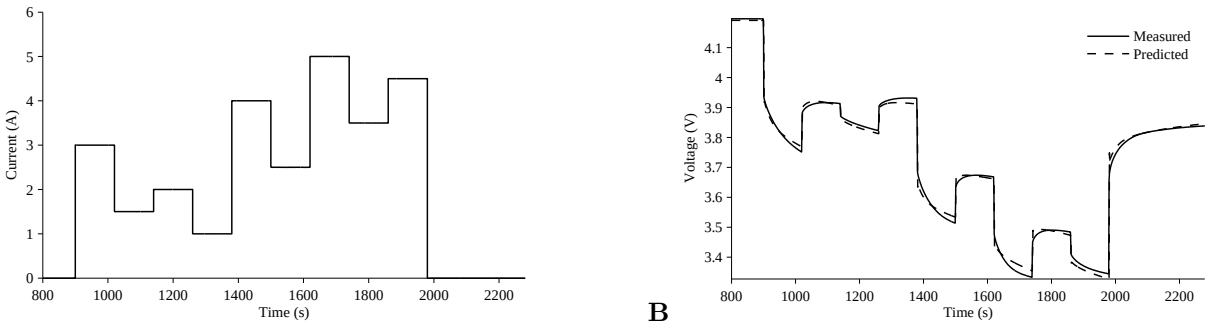
III.B. Battery Voltage

Battery potentials as seen in Figure 1 at the current collectors are described by several voltage terms. At the positive current collector is the equilibrium potential $V_{U,p}$. This voltage is then reduced by $V_{s,p}$, due to the solid-phase ohmic resistance, and $V_{\eta,p}$, the surface overpotential. The electrolyte ohmic resistance then causes another drop V_e. At the negative electrode, there is a drop $V_{\eta,n}$ due to the surface overpotential, and a drop $V_{s,n}$ due to the solid-phase resistance. The voltage drops again due to the equilibrium potential at the negative current collector $V_{U,n}$. We describe each of these voltage terms in turn. The total battery voltage potential can be expressed as below:

$$V = V_{U,p} - V_{U,n} - V_o - V_{\eta,p} - V_{\eta,n}. \tag{3}$$

Voltages in the battery are not observed to change instantaneously, i.e., the voltage change occurs smoothly. When discharge completes, for example, the voltage rises slowly as the surface layers move to the concentrations of the bulk volumes, as caused by diffusion. In addition to this, there are transients associated with V_o and the $V_{\eta,i}$ terms. To take this into account in a simple way, we compute voltage using

$$V = V_{U,p} - V_{U,n} - V'_o - V'_{\eta,p} - V'_{\eta,n}, \tag{4}$$



A B

Figure 2. Model validation for variable loading. A current over time. B measured and predicted variable loading discharge curves.

where

$$\dot{V}'_o = (V_o - V'_o)/\tau_o, \tag{5}$$

$$\dot{V}'_{\eta,p} = (V_{\eta,p} - V'_{\eta,p})/\tau_{\eta,p}, and \tag{6}$$

$$\dot{V}'_{\eta,n} = (V_{\eta,n} - V'_{\eta,n})/\tau_{\eta,n}, \tag{7}$$

and the τ parameters are empirical time constants.

The model contains as states $\mathbf{x}$, $q_{s,p}$, $q_{b,p}$, $q_{b,n}$, $q_{s,n}$, V'_o, $V'_{\eta,p}$, and $V'_{\eta,n}$. The single model output is V. Model validation for a variable loading scenario is shown in Fig. 2. The load changes every 2 minutes (Fig. 2A), resulting in corresponding changes in voltage. Fig. 2B shows that the voltage predictions are fairly accurate in response to changes in load. Some errors are still present that may possibly be accounted for by including temperature effects.

IV. Prognostics

In this section we discuss our developed battery prognosis framework following the general estimation-prediction methodology of model-based prognostics.[13–16] Similar approaches have been used for prognosis of pneumatic valves[17,18] and for Current/Pressure (I/P) Transducers (IPT).[19,20] Here, we only summarize the formulation of the prognostics problem, followed by a brief description of the estimation and prediction approach.

IV.A. Problem Formulation

The system model may be generally defined as

$$\mathbf{x}(k+1) = \mathbf{f}(k, \mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k), \mathbf{v}(k)), \text{and} \tag{8}$$

$$\mathbf{y}(k) = \mathbf{h}(k, \mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k), \mathbf{n}(k)), \tag{9}$$

where k is the discrete time variable, $\mathbf{x}(k) \in \mathbb{R}^{n_x}$ is the state vector, $\boldsymbol{\theta}(k) \in \mathbb{R}^{n_\theta}$ is the unknown parameter vector, $\mathbf{u}(k) \in \mathbb{R}^{n_u}$ is the input vector, $\mathbf{v}(k) \in \mathbb{R}^{n_v}$ is the process noise vector, $\mathbf{f}$ is the state equation, $\mathbf{y}(k) \in \mathbb{R}^{n_y}$ is the output vector, $\mathbf{n}(k) \in \mathbb{R}^{n_n}$ is the measurement noise vector, and $\mathbf{h}$ is the output equation.[a]

In prognostics, the occurrence of an event E is to be predicted, that is defined with respect to the states, parameters, and inputs of the system. The event is defined as the earliest instant that some event threshold $T_E : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_u} \to \mathbb{B}$, where $\mathbb{B} \triangleq \{0, 1\}$ changes from the value 0 to 1. That is, the time of the event k_E at some time of prediction k_P is defined as

$$k_E(k_P) \triangleq \inf\{k \in \mathbb{N} \colon k \geq k_P \wedge T_E(\mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k)) = 1\}. \tag{10}$$

The time remaining until that event, Δk_E, is defined as

$$\Delta k_E(k_P) \triangleq k_E(k_P) - k_P. \tag{11}$$

[a]Bold typeface denotes vectors, and n_a denotes the length of a vector $\mathbf{a}$.

For system health management, T_E is defined via a set of performance constraints that define what the acceptable states of the system are, based on $\mathbf{x}(k)$, $\boldsymbol{\theta}(k)$, and $\mathbf{u}(k)$.[16] For batteries, we are interested in predicting the end of discharge (EOD) time, i.e., the time at which the battery voltage will deplete below the voltage threshold V_{EOD}.

Models of the system components are constructed in this paradigm that capture both nominal behavior, as well as faulty behavior and damage progression. Using these models, observations can be mapped back to the health state of the system as represented in $\mathbf{x}$ and $\boldsymbol{\theta}$. An estimation algorithm, such as the Kalman filter (KF), unscented Kalman filter (UKF), or particle filter (PF), is used to solve these types of problems.[21] In this work an UKF based approach in implemented. This state-parameter estimate, along with a prediction of the future usage of the component, is used as input to a prediction algorithm that computes the time to EOD. The difference between EOD and current time is called the remaining useful life (RUL).[16,22]

IV.B. Prognostics Architecture

In our model-based prognostics architecture,[16] there are two sequential problems, (*i*) the *estimation* problem, which requires determining a joint state-parameter estimate $p(\mathbf{x}(k), \boldsymbol{\theta}(k)|\mathbf{y}(k_0{:}k))$ based on the history of observations up to time k, $\mathbf{y}(k_0{:}k)$, and (*ii*) the *prediction* problem, which determines at k_P, using $p(\mathbf{x}(k), \boldsymbol{\theta}(k)|\mathbf{y}(k_0{:}k))$, a probability distribution $p(k_E(k_P)|\mathbf{y}(k_0{:}k_P))$. The distribution for Δk_E can be trivially computed from $p(k_E(k_P)|\mathbf{y}(k_0{:}k_P))$ by subtracting k_P.

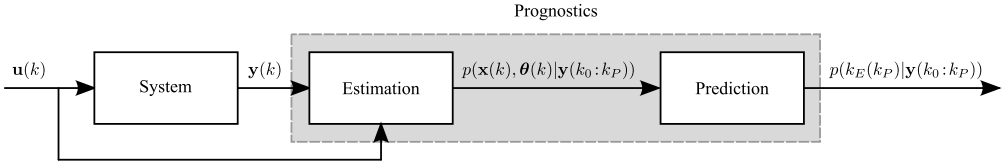


Figure 3. Prognostics architecture.[16]

The prognostics architecture is shown in Figure 3. In discrete time k, the system is provided with inputs $\mathbf{u}_k$ and provides measured outputs $\mathbf{y}_k$. The estimation module uses this information, along with the system model, to compute an estimate $p(\mathbf{x}(k), \boldsymbol{\theta}(k)|\mathbf{y}(k_0{:}k))$. The prediction module uses the joint state-parameter distribution and the system model, along with hypothesized future inputs, to compute the probability distribution $p(k_E(k_P)|\mathbf{y}(k_0{:}k_P))$ at given prediction times k_P.

IV.C. Estimation

A detailed electro-chemistry (EC) based physics model of component behavior is developed using nominal data from the testbed.[13] In this work the developed EC model is being implemented to estimate and predict health state of the battery. An unscented Kalman filter (UKF) is implemented to obtain the state estimate from the sensor measurements. Details of the implemented framework for battery modeling is discussed in Daigle & Kulkarni.[13]

IV.D. Prediction

Whereas the SOC value can be calculated directly from the current state, the calculation of EOL requires an estimate of the future load profile. With that information, a Monte Carlo forward integration is used to obtain the mean of EOL and its distribution. For an electric UAS, the power requirements are directly related to the planned future flight path. Here, we assume that the flight computer, the sensors, and the avionics is operated using a different battery, which is not considered here [b].

Our model-based prediction architecture, shown in Figure 4, extends the basic architecture from Figure 3 by an extended model-based prediction component which, given the planned future flight plan first estimates a load profile and then uses that to perform the prediction.

Flight plans for UAS are usually specified as a list of waypoints that the UAS visits one by one. Each waypoint can be seen as a four-dimensional point $\langle lat, lon, alt, v\rangle$ with latitude lat, longitude lon, altitude alt, and target (ground) speed v. For our simplified prognostics model, we make the assumption that the aircraft

[b] In general, the batteries for avionics and on-board computation should be monitored as well, as they might use up to 25% of the overall UAS power consumption.[23]

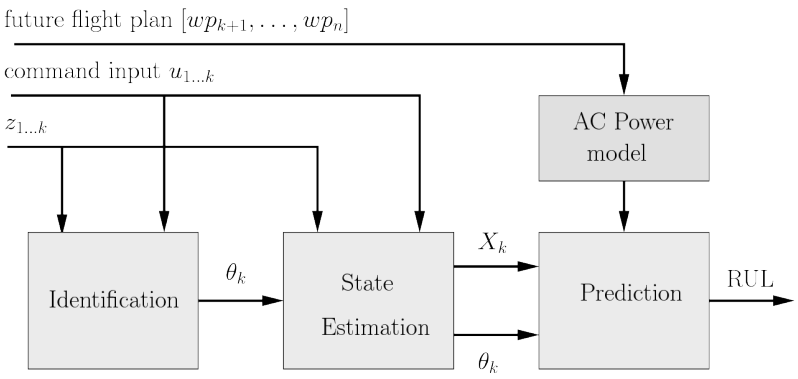


Figure 4. Prognostics architecture with identification, state estimation and prediction component that is driven by power requirements based on the future flight plan.

flies a straight line between two waypoints, i.e., it flies with a constant heading, vertical, and horizontal speed.

For this paper, we use a simplified model for a small electric fixed-wing aircraft. Then we can calculate the power requirements at each point in time. The total power requirements P can be given as[23]

$$P = \frac{gm}{\eta}\left(\frac{\mu(v \cdot a)}{g} + \frac{|v - v_W|}{C_{LD}} \times \frac{\cos\gamma + C_{LD}\sin\gamma}{\cos\phi}\right), \tag{12}$$

where v is the speed (vector) of the UAS, v_W the windspeed, a the acceleration, γ the climb angle, and ϕ the roll angle. C_{LD} is the aerodynamic lift-over-drag constant of the UAS, m its mass, and g the gravitational constant. The first summand of the formula describes the power requirements resulting from the dynamic forces, the second one the power needed to overcome drag. η summarizes the efficiency of propeller, motor, and ESC.

Figure 5 shows simulation results on a flight of a fixed-wing RC model aircraft (Edge R5,[5]) in the Xplane 11[c] simulator. The plot shows the various speeds, pitch and roll angle, and the actual and estimated power. For that aircraft, we use a fixed $C_{LD} = 4.3$, $\eta = 0.71$, and $\mu = 7$. Some deviations are visible upon quick change of throttle and for throttle settings close to zero.

Because of our simplifying assumptions, the total load $L_{wp_1 \to wp_2}$ can be easily calculated as

$$L_{wp_1 \to wp_2} = P \times t_{wp_1 \to wp_2}$$

where $t_{wp_1 \to wp_2}$ is the estimated flight time between the two waypoints.

Uncertainties and failures can be injected into this model easily. Most prominently, the wind speed v_w can play a major roll in power usage. Failures in the propulsion system can be modeled by modifying the parameter η.

V. The R2U2 Framework

R2U2 (Realizable, Responsive, and Unobtrusive Unit)[24–26] is an on-board monitoring system that has been developed to continuously monitor system and safety properties of the aircraft in flight. Health models within this framework[27,28] are defined using Metric Temporal Logic (MTL) and Mission-time Linear Temporal Logic (LTL)[25] for expressing temporal properties as well as Bayesian Networks (BN) for probabilistic and diagnostic reasoning. A signal processing unit (AT) reads in continuous sensor signals or information from the prognostics unit and performs filtering and discretization operations.

V.A. Temporal Logic Monitors

LTL and MTL formulas consist of propositional variables (which are usually comprised of discretized sensor values as produced by the AT component), the usual Boolean operators $\wedge$, $\vee$, $\neg$, or $\rightarrow$, and temporal operators. For formulas p, q, we have $\Box p$ (Always p), $\Diamond p$ (Eventually p), $\mathcal{X}p$ (NextTime p), $p\,\mathcal{U}q$ (p Until q),

[c] www.x-plane.com

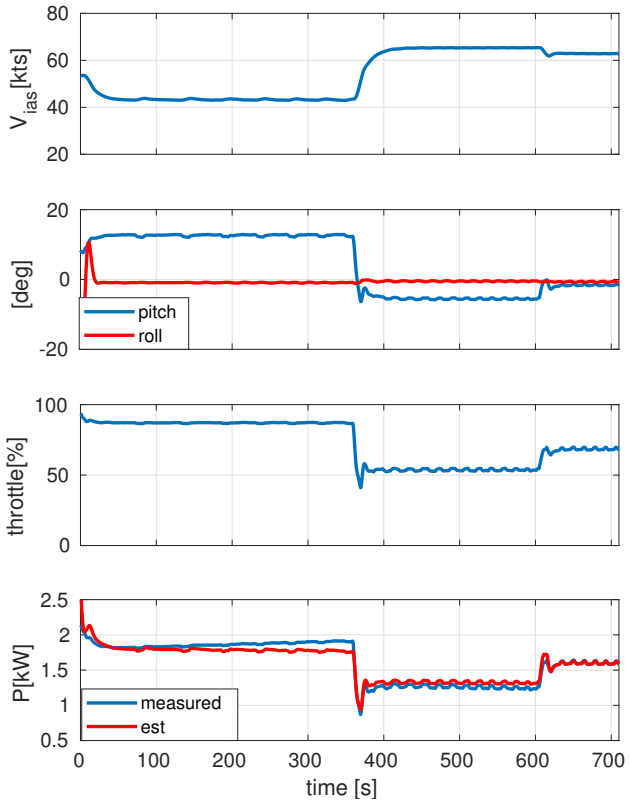


Figure 5. Power profile of a simulated flight with Edge R5 in XPlane 11 simulator: power consumption as calculated by the simulator and estimated by our model.

and $p\,\mathcal{R}q$ (p RELEASES q). For MTL, each of the temporal operators is accompanied by an upper and lower time bound that express the time period during which the operator must hold. For example, $\Box_{[2\,\text{sec},6\,\text{sec}]}p$ means that p must be true at all times between 2 sec and 2 sec along the time line. $\Diamond_{[0\,\text{sec},7\,\text{sec}]}p$ means that p must be true at least once in the interval 0 sec, 7 sec. Times in R2U2 are discreet and depend on th execution rate of the R2U2 engine, which is usually running at 10 Hz. For a detailed definition and semantics of the temporal operators see.[25] Properties regarding the (electric) propulsion system and the battery status can be assigned into three different categories, which we will briefly discuss in the following.

V.A.1. Instantaneous Safety Properties

Such safety properties must hold during the entire flight and can be evaluated instantaneously. A typical example, $\Box(U_{batt} > 13.5\,\text{V})$, indicates that the battery voltage must never drop below 13.5 V. Another rule might say that the SOC of the battery is at least 50%: $\Box(\mu^{SOC} > 50\%)$, where μ^{SOC} indicates the mean value of SOC as returned by the prognostics engine.

A large number of safety rules for the entire propulsion system and even the entire aircraft behavior can be formulated in a similar way. R2U2 continuously checks these properties and indicates violations of any of these properties. That information can be directly passed to the pilot or auto-pilot or may be used by the Bayesian reasoner[27] to perform root cause analysis.

V.A.2. Temporal Safety Properties

These safety properties specify temporal relation ships that must hold for a safe flight. For example, the battery-related property

$$\Box((\mu^{SOC} < 50\%) \rightarrow (I_{batt} > 30\,\text{A})\mathcal{U}_{[0\,\text{sec},29\,\text{sec}]}(I_{batt} \leq 30\,\text{A}))$$

is violated, if, when the battery is weak (its mean state of charge μ^{SOC} is less than 50%), more than 30 A is drawn for more than 30 consecutive seconds. That rule could help to avoid a dangerous situation, in

which a (weak) battery might heat up. A number of examples belonging to this category can be found in literature.[11,24]

V.A.3. Prognostics-based Safety Properties

There is a number of safety properties that define flight safety with respect to the future flight path of the aircraft. For example, the current flight plan requires the aircraft to fly over a high mountain. Obviously the climb to the top uses up considerable battery power, so the pilot (or autopilot) must ensure that during the 10-minute climb, the battery level never goes below 30%. Such a property can be easily specified in temporal logic within our R2U2 framework:

$$\square(\texttt{climb-mountain} \rightarrow \square_{[0\,\mathrm{min},9\,\mathrm{min}]}(\mu^{SOC} > 30\%).$$

There is only one catch: R2U2 is no magic crystal ball, which can look into the future. Thus the above formula can only be valuated after 10 minutes; prior to that time, R2U2 returns `maybe`. This is, of course, not helpful if the pilot or auto-pilot has to make the decision if to fly over the mountain or not *prior* to even attempting the climb.

We therefore integrate the prognostics engine into the R2U2 framework. We now can formulate safety properties that directly access prognostics information. The above safety property now would be simply formulated as:

$$\square(\texttt{climb-mountain} \rightarrow \mathit{RUL}(\texttt{climb}) > 10\,\mathrm{min}),$$

where $\mathit{RUL}(\texttt{climb})$ is the rest of useful life estimate, after the flight plan, given in the argument has been flown. This formula can be evaluated immediately and can indicate to the pilot if the climb can be attempted safely or not. In Section VI, we discuss such properties in more detail.

VI. Case Studies

VI.A. SOC Estimate for an Octo-copter

The prognostics architecture and R2U2 was implemented as part of the AOS system[29] and test-flown successfully on an X8+ octo-copter at the NASA Ames Research Center. For safety reasons, all flights had been executed while the UAS was tethered. R2U2 and the prognostics engine monitor the UAS system, flight rules, and the battery with an update rate of 0.5 Hz. After manual takeoff, the aircraft was directed to approach the traffic pattern of a scaled-down version of a traffic pattern near an airport. After obtaining ATC clearance to enter the traffic pattern, the UAS flies along the traffic pattern. However, no clearance for landing is given, so the UAS has to fly go-arounds. During the entire flight, the battery (voltage and current) and numerous signals are monitored. R2U2 checks for nominal battery voltage (`BATT_VOLTAGE_NOM`) as well as for battery voltages that are sufficient for a climb (`BATT_VOLTAGE_CLIMB`) as well as other battery-related signals. Figure 6A shows the three-valued outputs of the R2U2 observers, shown in 0.5Hz timesteps.

During this test flight, the UAS enters the traffic pattern at around $t = 300s$ and flies 4 circles around the traffic pattern (as indicated by `mitigation_T0`). The battery is being discharged and at around $t = 500s$, the voltage sometimes drops below the minimal level suggested for a climb (`BATT_VOLTAGE_CLIMB`). At $t = 600s$ the battery voltage drops below the abort threshold of $U_{batt} < 14.2V$ for more than 10 consecutive seconds, causing the signal `mitigation_T1` to become true. On-board logic then requests an immediate clear-to-land from ATC, which is indicated by the R2U2 output `ATC_CLR_LANDING`. This causes the autopilot to carry out an immediate landing, which ends the scenario.

Figure 6B shows measured battery voltage and current as well as the estimated SOC in % during the flight. The bottom panel shows the estimated RUL for a predicted constant current draw of $\bar{I} = 20A$. Since this scenario does not involve any strong climbs, the estimation of RUL with a constant load seems to be appropriate. Uncertainties in the current draw can be modeled in our architecture by carrying out the RUL estimation for samples from a Gaussian distributed load current $I = \mathcal{N})(\bar{I}, \sigma)$. Figure 7 shows how the probability density function (PDF) for $\bar{I} = 20A$ and $\sigma = 4$ develops during the flight time. As expected, the mean of RUL is a straight line as in Figure 6B and the variance of RUL decreases toward the end of the flight experiment.

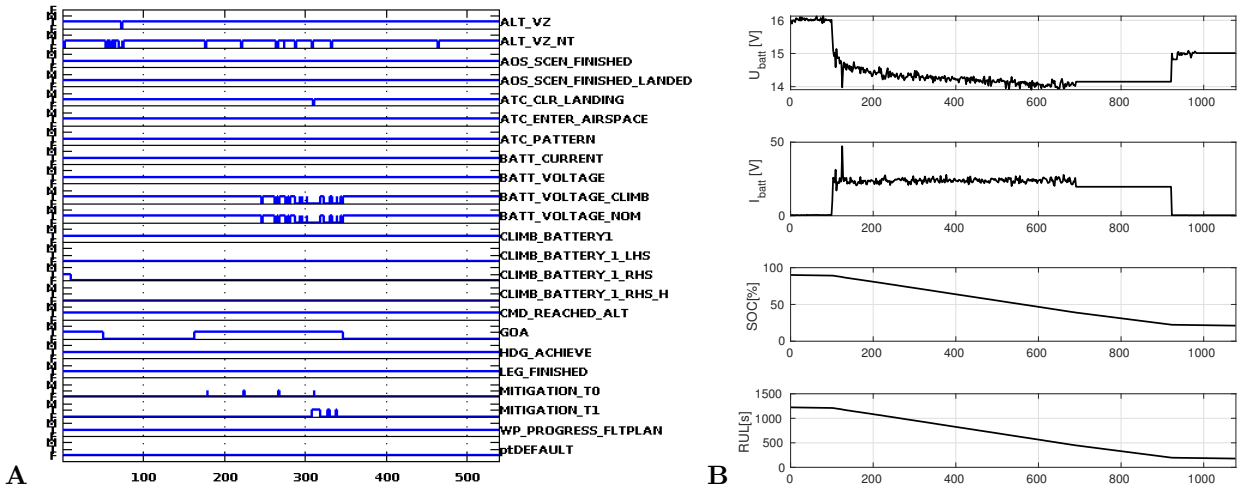


Figure 6. A: Outputs of R2U2 for several safety properties over flight time (in 2s intervals). B: Measured battery voltage and current, estimated SOC and RUL for a constant load profile.

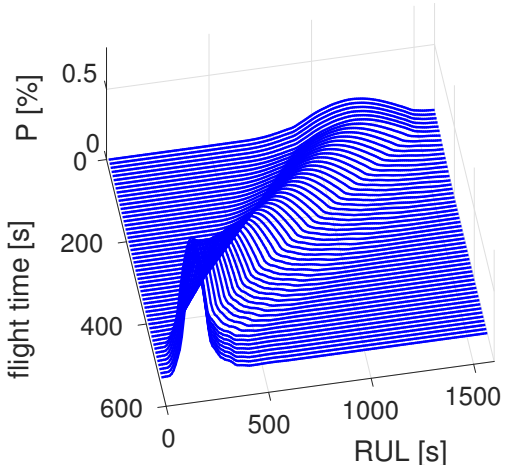


Figure 7. Probability distribution for RUL over flight time for scenario shown in Figure 6B.

VI.B. RUL Prognosis for a fixed-wing electrical Aircraft

For this simulation case study, we use a simulation model of the Langley Edge R5 aircraft.[5] This aircraft is a large RC-style aircraft with 6 ft wingspan and a take-off weight of approximately 18 kg. This aircraft is in active use at NASA LaRC for UAS-related research. An existing model for the dynamics and the battery of the aircraft made it a suitable candidate for this case study.

In an UAM (Urban Area Mobility) scenario, the aircraft must find as suitable trajectory between points A and B. Such a trajectory must obey numerous constraints (restricted zones, flight time, other aircraft, weather, etc), one of the most important ones concern the availability of enough battery power. In our simple demonstration scenario, the shortest direct path is flying over a mountain, a distance of approximately 26.5 km, which requires a strong climb by 1400 m. An alternative trajectory around the mountain does not require any climbs/descends, but is substantially longer (32 km). Here, we analyze (a) flying over the mountain, (b) going around the mountain at level altitude with high speed (60 kts), and (c) going around the mountain with an energy-conserving 40 kts.

While the aircraft is in the air, R2U2 and the prognostics engine continuously monitor the battery of the aircraft and calculate SOC and RUL for existing flight plan. If there is an opportunity or need to evaluate a different future flight plan (e.g., to slowly go around the mountain), PDFs for the RUL of the various trajectories are calculated on-board.

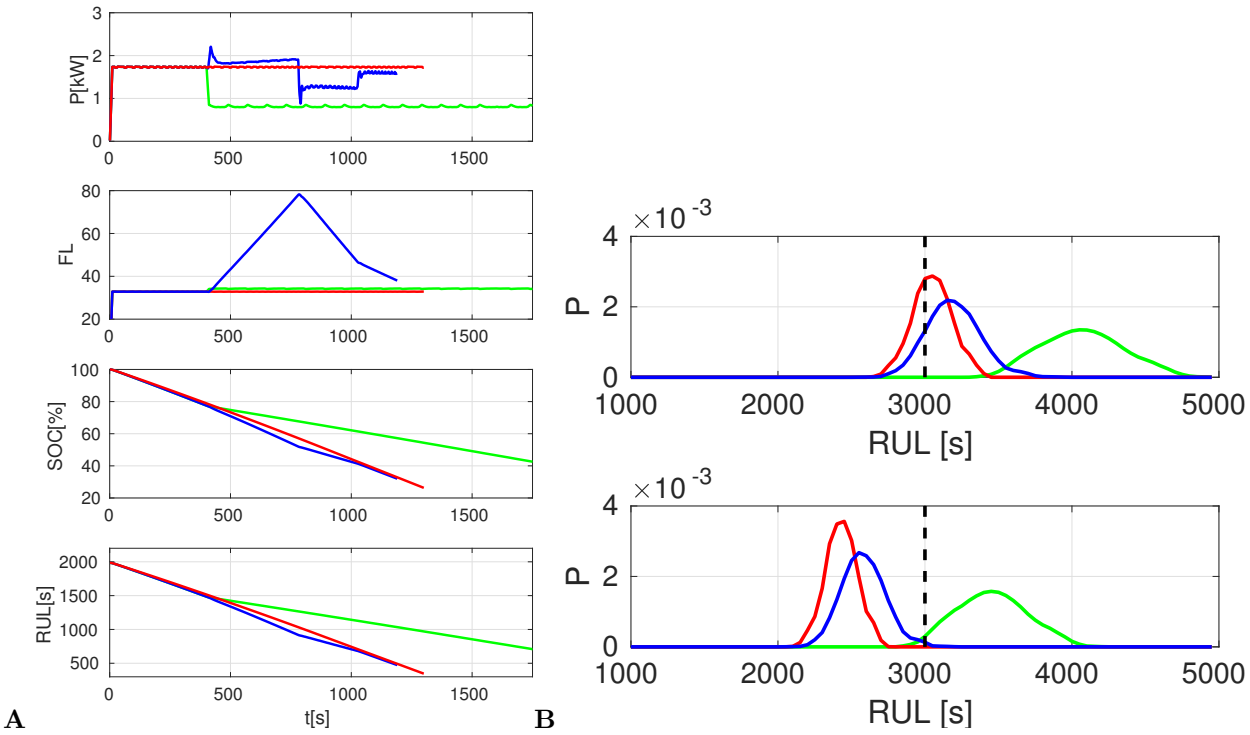


Figure 8. A: power requirements, altitudes, SOC, and RUL with constant power use for trajectory variants (a)—blue, (b)—red, and (c)—green.B: Probability densities for RUL at the beginning of the flight $t = 0s$ and at the decision point $t = 400s$ with $\sigma_{pwr} = 0.1$.

For these flight-plans, the power consumption is being calculated according to Equation (12) and the desired trajectory. The RUL calculation during the flight uses the actual battery state into account, which is updated using measured power consumption of the aircraft. Figure 8A shows the actual power requirements for the trajectories (a), (b), and (c), and the altitude profile in flight levels (FL). For the first 400 seconds all trajectories are the same. Obviously, the higher power consumption of (a) and (b) cause a lower SOC, which goes below 50%. Note that the mountain route and fast flight around the mountain reaches the destination earlier; hence the lines are shorter. If we calculate the RUL according the to high power usage similar to that in the initial flight segment, the faster, but power-hungrier variants end up with a shorter RUL.

If we require a minimum of at least 3000s RUL at the destination, both SOC and RUL estimation do not allow help us in the early stages of the flight to make a decision on which route can actually be flown. Only some time after our decision point at $t = 400$, we could see that only the slow route around the mountain allows for enough safety margin. We therefore perform RUL predictions using the variants of the intended flight path. In order to accommodate uncertainties, which might, for example, arise due to weather, we draw the power requirements from a Gaussian distribution. If current wind-speed and direction were available, an even more accurate prediction of RUL would be possible.

Figure 8B shows the resulting RUL distribution at the beginning of the flight ($t = 0$). At that point in time, all three variants seem to be possible, albeit variant (b) is risky, when we require a minimum RUL of 3000s (top panel). At $t = 400s$ (just before the decision has to be made), the situation looks differently: variants (a) and (b) are not safe, only the slow flight around the mountain obeys our safety requirements,

VII. Conclusions and Future Work

In this paper we presented the integration of prognostics reasoning of battery health using SOC and RUL into an overall R2U2 temporal and Bayesian health management framework. An UKF-based prognostics engine is being implemented to estimate SOC and RUL for battery used in an electric UAV. The prognostic results are being used by the R2U2 monitor to take certain decisions based on the power profiles for different flights paths. On-board temporal logic monitors are implemented to asses safety properties which are active

during the entire flight path. Thus, based on the state of health estimates and set temporal safety properties decisions are taken to change the flight path as required to complete the mission safely. In the first case the framework is demonstrated on an octo-copter simulation wherein on reaching the SOC cut-off threshold, the on-board logic requests an clear-to-land from the ATC to make a safe landing without obstructing any other vehicles around. In the second example the implemented framework on a fixed wing vehicle is able to take decision to choose a path which requires minimum amount of energy based on future flight paths and reach its destination with sufficient amount of SOC in the battery packs. Thus the framework presented in this work ensures the mission is successful in spite of change in the flight path based on the battery state of health and energy consumption estimates.

Acknowledgements

This work was funded by NASA System Wide Safety (SWS) project under the Airspace Operations and Safety (AOSP) program, and the Autonomous Operating System (AOS) project under the Convergent Aeronautics Solutions (CAS) program within the NASA Aeronautics Research Mission Directorate (ARMD).

References

[1]Bundesanstalt für Verkehr, "Untersuchungsbericht Flugunfall mit dem Motorflug zeug Type Airbus A310 am 12. Juli 2000 am Flughafen Wien-Schwechat, Niederösterr eich," Tech. rep., Bundesanstalt für Verkehr, Austria, 2006.

[2]Bole, B., Teubert, C., Chi, Q. C., Edward, H., Vazquez, S., Goebel, K., and Vachtsevanos, G., "SIL/HIL replication of electric aircraft powertrain dynamics and inner-loop control for V&V of system health management routines," *Annual Conference of the Prognostics and Health Management Society*, 2013.

[3]Quach, C. C., Bole, B., Hogge, E., Vazquez, S., Daigle, M., Celaya, J., and Goebel, K., "Battery Charge Depletion Prediction on an Electric Aircraft," *Annual Conference of the Prognostics and Health Management Society*, 2013.

[4]Hogge, E., Bole, B., Vazquez, S., Kulkarni, C., Strom, T., Hill, B., Smalling, K., and Quach, C., "Verification of Prognostic Algorithms to Predict Remaining Flying Time for Electric Unmanned Vehicles," *International Journal of Prognostics and Health Management, ISSN 2153-2648, 2018 021*, 2018.

[5]Hogge, E., Bole, B., Vazquez, S., Strom, T., Hill, B., Smalling, K., and Quach, C., "Verification of a Remaining Flying Time Prediction System for Small Electric Aircraft," *Annual Conference of the Prognostics and Health Management Society*, 2015.

[6]Drusinsky, D., "The Temporal Rover and the ATG Rover," *SPIN*, Vol. 1885 of *Lecture Notes in Computer Science*, Springer Verlag, 2000, pp. 323–330.

[7]Tabakov, D., Rozier, K. Y., and Vardi, M. Y., "Optimized Temporal Monitors for SystemC," *Formal Methods in System Design*, Vol. 41, No. 3, 2012, pp. 236–268.

[8]Pike, L., Wegmann, N., Niller, S., and Goodloe, A., "Copilot: monitoring embedded systems," *Innovations in Systems and Software Engineering*, Vol. 9, No. 4, 2013, pp. 235–255.

[9]Pellizzoni, R., Meredith, P., Caccamo, M., and Rosu, G., "Hardware Runtime Monitoring for Dependable COTS-Based Real-Time Embedded Systems," *RTSS*, 2008, pp. 481–491.

[10]Meredith, P. O., Jin, D., Griffith, D., Chen, F., and Roşu, G., "An overview of the MOP runtime verification framework," *International Journal on Software Tools for Technology Transfer*, Vol. 14, No. 3, 2012, pp. 249–289.

[11]Schumann, J., Roychoudhury, I., and Kulkarni, C., "Diagnostic Reasoning using Prognostic Information for Unmanned Aerial Systems," *PHM 2015*, 2015.

[12]Rahn, C. D. and Wang, C.-Y., *Battery Systems Engineering*, Wiley, 2013.

[13]Daigle, M. and Kulkarni, C., "Electrochemistry-based Battery Modeling for Prognostics," *Annual Conference of the Prognostics and Health Management Society 2013*, Oct. 2013, pp. 249–261.

[14]Luo, J., Pattipati, K. R., Qiao, L., and Chigusa, S., "Model-based prognostic techniques applied to a suspension system," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol. 38, No. 5, Sept. 2008, pp. 1156 –1168.

[15]Orchard, M. and Vachtsevanos, G., "A Particle Filtering Approach for On-Line Fault Diagnosis and Failure Prognosis," *Transactions of the Institute of Measurement and Control*, , No. 3-4, June 2009, pp. 221–246.

[16]Daigle, M. and Goebel, K., "Model-based Prognostics with Concurrent Damage Progression Processes," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 43, No. 4, May 2013, pp. 535–546.

[17]Daigle, M., Kulkarni, C., and Gorospe, G., "Application of Model-based Prognostics to a Pneumatic Valves Testbed," *Proceedings of the 2014 IEEE Aerospace Conference*, March 2014.

[18]Kulkarni, C., Daigle, M., Gorospe, G., and Goebel, G., "Validation of Model-Based Prognostics for Pneumatic Valves in a Demonstration Testbed," *Annual Conference of the Prognostics and Health Management Society 2014*, Oct. 2014.

[19]Teubert, C. and Daigle, M., "I/P Transducer Application of Model-Based Wear Detection and Estimation using Steady State Conditions," *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2013*, Oct. 2013, pp. 134–140.

[20]Teubert, C. and Daigle, M., "Current/Pressure Transducer Application of Model-Based Prognostics using Steady State Conditions," *2014 IEEE Aerospace Conference*, March 2014.

[21]Daigle, M., Saha, B., and Goebel, K., "A comparison of filter-based approaches for model-based prognostics," *Proceedings of the 2012 IEEE Aerospace Conference*, March 2012.

[22]Daigle, M., Saxena, A., and Goebel, K., "An Efficient Deterministic Approach to Model-based Prediction Uncertainty Estimation," *Annual Conference of the Prognostics and Health Management Society*, Sept. 2012, pp. 326–335.

[23]Caccamo, M., "Power-Aware Emulation Environment for long-endurance Solar UAVs," 2017.

[24]Rozier, K. Y. and Schumann, J., "R2U2: Tool Overview," *RV-CuBES 2017. An International Workshop on Competitions, Usability, Benchmarks, Evaluation, and Standardisation for Runtime Verification Tools, September 15, 2017, Seattle, WA, USA*, 2017, pp. 138–156.

[25]Reinbacher, T., Rozier, K. Y., and Schumann, J., "Temporal-Logic Based Runtime Observer Pairs for System Health Management of Real-Time Systems," *Tools and Algorithms for the Construction and Analysis of Systems - 20th International Conference, TACAS 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014. Proceedings*, edited by E. Ábrahám and K. Havelund, Vol. 8413 of *Lecture Notes in Computer Science*, Springer, 2014, pp. 357–372.

[26]Geist, J., Rozier, K. Y., and Schumann, J., "Runtime Observer Pairs and Bayesian Network Reasoners On-board FPGAs: Flight-Certifiable System Health Management for Embedded Systems," *Runtime Verification - 5th International Conference, RV 2014, Toronto, ON, Canada, September 22-25, 2014. Proceedings*, edited by B. Bonakdarpour and S. A. Smolka, Vol. 8734 of *Lecture Notes in Computer Science*, Springer, 2014, pp. 215–230.

[27]Schumann, J., Rozier, K. Y., Reinbacher, T., Mengshoel, O. J., Mbaya, T., and Ippolito, C., "Towards Real-time, On-board, Hardware-supported Sensor and Software Health Management for Unmanned Aerial Systems," *Proceedings of the 2013 Annual Conference of the Prognostics and Health Management Society (PHM2013)*, October 2013, pp. 381–401.

[28]Schumann, J., Rozier, K. Y., Reinbacher, T., Mengshoel, O. J., Mbaya, T., and Ippolito, C., "Towards Real-time, On-board, Hardware-supported Sensor and Software Health Management for Unmanned Aerial Systems," *International Journal of Prognostics and Health Management*, October 2015, pp. to appear.

[29]Lowry, M., Bajwa, A. R., Pressburger, T., Sweet, A., Dalal, M., Fry, C., Schumann, J., Dahl, D., Karsa, G., and Mahadevan, N., "Design Considerations for a Variable Autonomy Exeuctive for UAS in the NAS," *2018 AIAA Information Systems-AIAA Infotech @ Aerospace*, 2018.